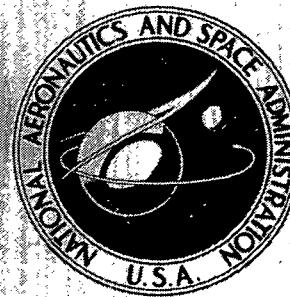


N72-28890

NASA TECHNICAL
MEMORANDUM

NASA TM X-2592



CASE FILE
COPY

COMPUTER PROGRAM FOR INVESTIGATING
EFFECTS OF NONLINEAR SUSPENSION-SYSTEM
ELASTIC PROPERTIES ON PARACHUTE
INFLATION LOADS AND MOTIONS

by Lamont R. Poole

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1972

NASA TM X-2592

1. Report No. NASA TM X-2592	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle COMPUTER PROGRAM FOR INVESTIGATING EFFECTS OF NONLINEAR SUSPENSION-SYSTEM ELASTIC PROPERTIES ON PARACHUTE INFLATION LOADS AND MOTIONS		5. Report Date August 1972	
7. Author(s) Lamont R. Poole		6. Performing Organization Code	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		8. Performing Organization Report No. L-8390	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		10. Work Unit No. 815-20-09-03	
15. Supplementary Notes		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
		14. Sponsoring Agency Code	
16. Abstract This paper presents a computer program by which the effects of nonlinear suspension-system elastic characteristics on parachute inflation loads and motions can be investigated. A mathematical elastic model of suspension-system geometry is coupled to the planar equations of motion of a general vehicle and canopy. Canopy geometry and aerodynamic drag characteristics and suspension-system elastic properties are tabular inputs. The equations of motion are numerically integrated by use of an equivalent fifth-order Runge-Kutta technique.			
17. Key Words (Suggested by Author(s)) Parachutes Parachute inflation Suspension-system elasticity		18. Distribution Statement Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 44	22. Price* \$3.00

COMPUTER PROGRAM FOR INVESTIGATING EFFECTS OF
NONLINEAR SUSPENSION-SYSTEM ELASTIC PROPERTIES ON
PARACHUTE INFLATION LOADS AND MOTIONS

By Lamont R. Poole
Langley Research Center

SUMMARY

A computer program has been developed to investigate the effects of nonlinear suspension-system elastic properties on longitudinal vehicle and canopy motions and tension loads experienced during the parachute inflation process. A mathematical elastic model of the suspension system is coupled to the planar equations of motion of a general vehicle and parachute canopy. Suspension-system elastic properties and histories of canopy drag parameter (product of drag coefficient and reference area), projected-area ratio (ratio of instantaneous projected area to projected area at full inflation), and added mass and its rate of change are input through the use of tables. The differential equations of motion of the system are integrated by use of an equivalent fifth-order Runge-Kutta technique.

INTRODUCTION

The successful operation of a parachute consists of two main phases: a deployment phase, in which the suspension lines and canopy are extended downstream of the towing vehicle, and an inflation phase, in which a net flow of air into the canopy alters the shape of the canopy until a final state of full inflation is reached. It is during this inflation phase that large forces are exerted on the towing vehicle by the parachute.

It was shown in reference 1 that the elasticity of the parachute suspension lines can have an important effect on the loads exerted on the towing vehicle during the inflation process. In reference 1 the suspension lines were modeled as a linear massless spring with no viscous damping, and the analysis was limited to the case of infinite-mass (constant dynamic pressure) parachute deployment conditions. It was shown in reference 2 that a massless-spring elasticity model provides a good approximation of the dynamic response of the suspension lines when loading conditions exist under which suspension-line wave mechanics can be ignored. Such conditions are normally in effect during parachute inflation. The purpose of the present paper is to describe a computer program, PLAPIP (Planar Parachute Inflation Program), by which the analysis of parachute loads

can be extended to include the effects of nonlinear spring characteristics, viscous damping, and trajectory gradients on the loads transmitted to the towing vehicle.

The function of program PLAPIP is to solve the differential equations governing the planar motion of a general towing vehicle and the linear motion of an inflating parachute canopy relative to the vehicle. The two bodies are coupled by a two-component mathematical elastic model of the suspension system, which involves an additional equation governing the position of the juncture of the two suspension-system components. The differential equations are solved by use of an equivalent fifth-order Runge-Kutta integration technique.

SYMBOLS

A	reference area, meters ²
C	damping coefficient per member, newton-seconds
C _D	drag coefficient, $\frac{\text{Drag}}{qA}$
g	acceleration due to gravity, meters/second ²
h	altitude, meters
K' _{sec}	specific secant modulus per member, newtons
L	unstressed length, meters
m	mass, kilograms
n _{bl}	number of bridle legs
n _{sl}	number of suspension lines
q	dynamic pressure, newtons/meter ²
T _s	component, along the central axis, of sum of tensions in all suspension lines, newtons
T _b	component, along the central axis, of sum of tensions in all bridle legs, newtons

t	time, seconds
v	velocity, meters/second
x	distance measured relative to base of towing vehicle, positive rearward, meters
β	angle between central axis of parachute and any suspension line, degrees
γ	vehicle flight-path angle, degrees
ξ	angle between central axis of parachute and any bridle leg, degrees
ρ	density, kilograms/meter ³

Subscripts:

a	added (enclosed and apparent)
bl	bridle leg
c	canopy
cs	canopy skirt
o	initial conditions
p	confluence point
sl	suspension line
v	vehicle
∞	free stream

Dots over symbols denote time derivatives.

PROBLEM DESCRIPTION

The problem considered in this paper is that of calculating the vehicle loads and motions which result from the inflation of a parachute attached to the vehicle by an elastic suspension system. It is assumed that histories of canopy projected-area ratio (ratio of projected area at any time to projected area at full inflation), drag parameter (product of drag coefficient and reference area), and added mass and its rate of change can be supplied from previous flight test data or auxiliary computer programs. The force transmitted to the towing vehicle is calculated by using elastic characteristics of the suspension-system components and a mathematical model of the geometry of the system. Vehicle trajectory and motion of the parachute canopy relative to the vehicle are determined.

For the purpose of this analysis, the vehicle and canopy are considered to be mass particles, with the mass of the canopy particle varying because of changes in the air mass enclosed by the canopy during inflation and the "apparent" mass effect of changes in the kinetic energy of the air surrounding the canopy (ref. 3, pp. 153-155). The surface of the planet is considered to be flat, and surface-relative accelerations are considered to be inertial. The motion of the towing vehicle is restricted to a vertical plane, and the motion of the canopy relative to the towing vehicle is restricted to occur along the central axis of the parachute. The central axis is considered to be parallel to the relative-wind velocity vector of the vehicle.

Equations of Motion

The forces affecting the motion of the towing vehicle and parachute canopy are shown in figure 1.

The motion of the towing vehicle is influenced by aerodynamic drag, tension in the parachute bridle, and gravitational attraction. The acceleration of the vehicle in the direction of its velocity vector is given by

$$\dot{v}_v = - \left[\frac{(C_D A)_v q_\infty + T_b}{m_v} + g \sin \gamma \right] \quad (1)$$

where $q_\infty = \frac{1}{2} \rho_\infty v_v^2$.

In order to specify fully the surface-relative planar motion of the vehicle, the following trajectory equations are required:

$$\dot{h}_v = v_v \sin \gamma \quad (2)$$

$$\dot{\gamma} = \frac{-g \cos \gamma}{v_v} \quad (3)$$

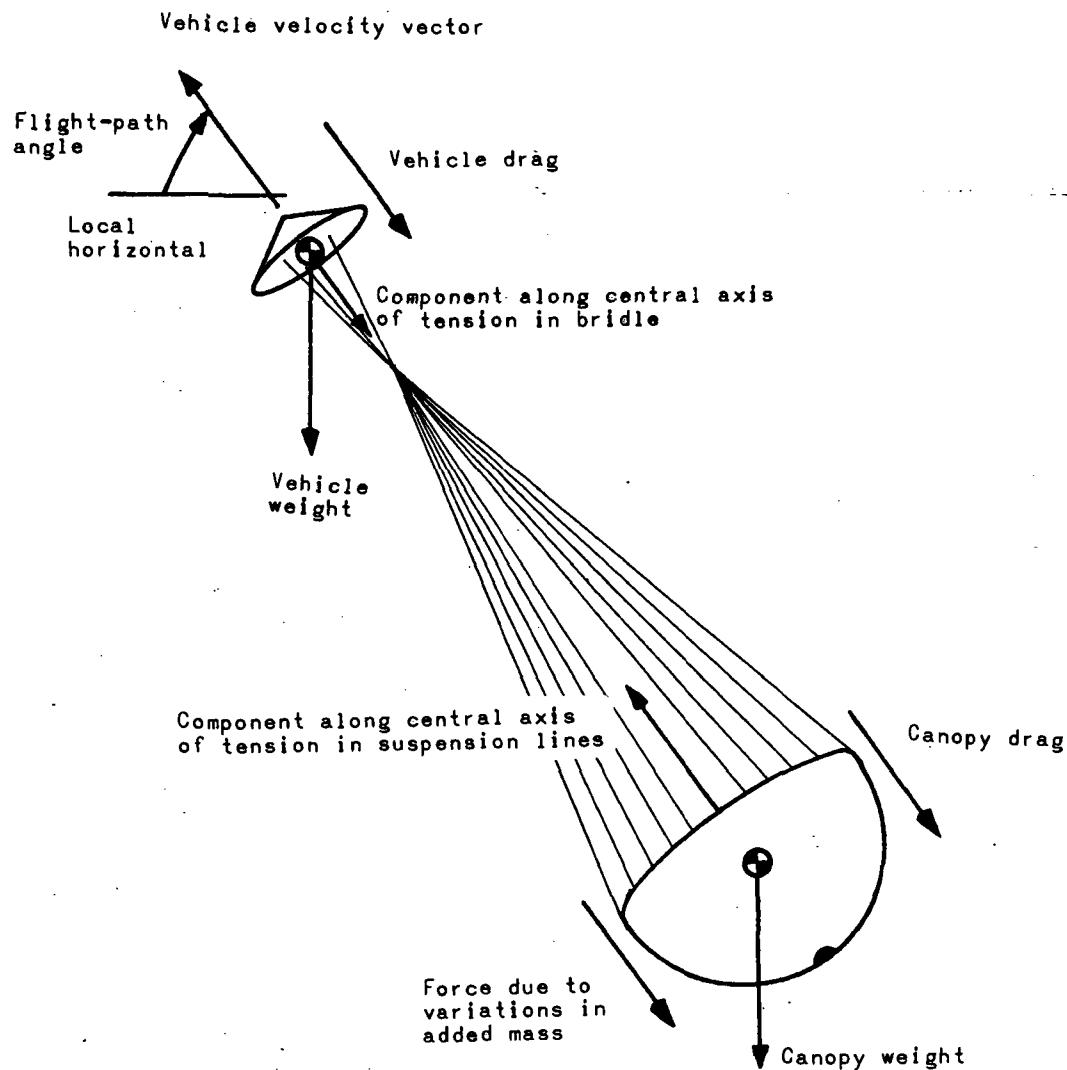


Figure 1.- Forces affecting motion of vehicle and canopy.

The motion of the inflating canopy is influenced by aerodynamic drag force ($C_{DA}q$), tension in the parachute suspension lines, a force due to changes in the added mass, and gravitational attraction. The acceleration of the canopy in the direction of the vehicle velocity vector is given by

$$\dot{v}_c = \frac{T_s - (C_{DA})_c q_c - m_a v_c - m_c g \sin \gamma}{m_c + m_a} \quad (4)$$

where m_a represents the added (enclosed and apparent) mass, and q_c is assumed to be equal to $\frac{1}{2} \rho_\infty v_c^2$.

In addition, the acceleration of the canopy relative to the vehicle can be written as

$$\ddot{x}_c = \dot{v}_v - \dot{v}_c \quad (5)$$

Mathematical Elastic Models of Suspension System

In order to couple the equations of motion of the towing vehicle and inflating canopy, it is necessary to formulate a relationship between the tension in the bridle, the tension in the suspension lines, and the motion of the canopy relative to the vehicle. By considering a particular suspension-system configuration and particular elastic characteristics, an equation can be formulated which defines the geometry of the suspension system and thus couples the equations of motion of the vehicle and canopy.

For this analysis the suspension system will be considered as two sets of components: suspension lines and a multilegged bridle, which are joined at a common confluence point. The suspension system is assumed to have negligible mass and negligible aerodynamic drag and is restricted to be under tension throughout the inflation process. The configuration for a fully inflated parachute, assuming axisymmetric inflation, is shown in figure 2.

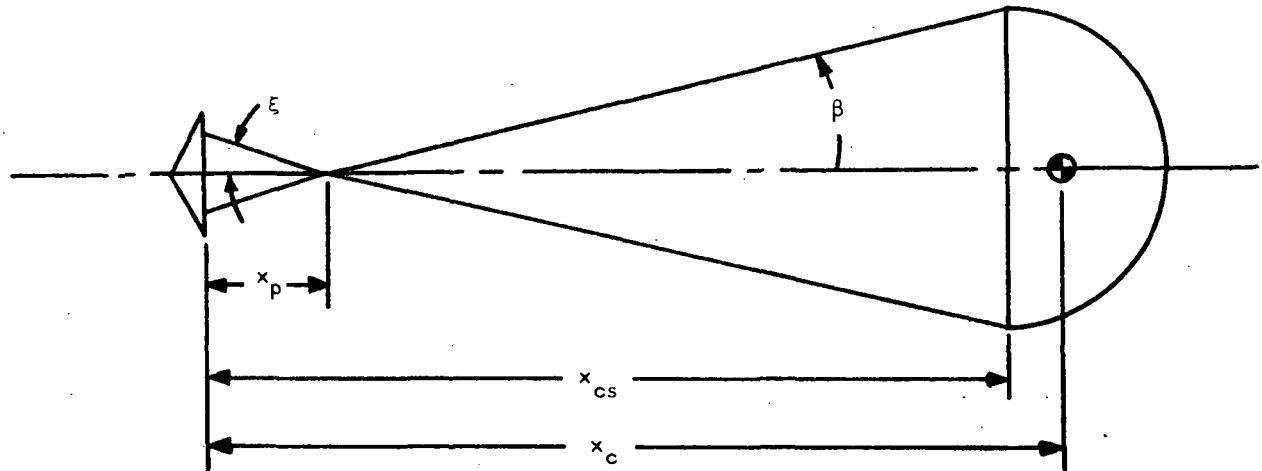


Figure 2.- Assumed configuration of suspension system.

By using the geometry shown in the figure and assuming uniform elastic properties, an equation defining the component along the central axis of the sum of tensions in all the suspension lines can be written as a function of the suspension-line strain and strain rate as follows:

$$T_s = n_{sl} \cos \beta \left\{ K'_{sec,sl} \left[\frac{(x_{cs} - x_p) \sec \beta}{L_{sl}} - 1 \right] + C_{sl} \frac{(\dot{x}_c - \dot{x}_p) \sec \beta}{L_{sl}} \right\} \quad (6)$$

(Note that $\dot{x}_c = \dot{x}_{cs}$.) The secant modulus $K'_{sec,sl}$ is a function of the strain, and the damping coefficient C_{sl} can be a function of both the strain and the strain rate.

Analogously, an equation defining the component along the central axis of the sum of the tensions in the bridle legs can be written as

$$T_b = n_{bl} \cos \xi \left[K'_{sec,bl} \left(\frac{x_p \sec \xi}{L_{bl}} - 1 \right) + C_{bl} \frac{\dot{x}_p \sec \xi}{L_{bl}} \right] \quad (7)$$

Under the assumption of negligible mass for the suspension system, the tension at any particular time is constant throughout the length of the suspension lines and bridle ($T_s = T_b$). Thus, by combining equations (6) and (7), an expression for calculating the rate of change of the relative position of the confluence point can be found, that is,

$$\dot{x}_p = \frac{\frac{n_{sl}}{L_{sl}} \left[C_{sl} \dot{x}_c + K'_{sec,sl} (x_{cs} - x_p - L_{sl} \cos \beta) \right] - \frac{n_{bl} K'_{sec,bl} (x_p - L_{bl} \cos \xi)}{L_{bl}}}{\frac{n_{sl} C_{sl}}{L_{sl}} + \frac{n_{bl} C_{bl}}{L_{bl}}} \quad (8)$$

This equation can be integrated to give the position of the confluence point relative to the towing vehicle.

If suspension-system viscous damping is negligible ($C_{sl} = C_{bl} = 0$), equations (6) and (7) can be combined to form an algebraic equation defining the relative position of the confluence point, that is,

$$x_p = \frac{\frac{n_{sl} K'_{sec,sl}}{L_{sl}} \left(\frac{x_{cs}}{L_{sl}} - \cos \beta \right) + n_{bl} K'_{sec,bl} \cos \xi}{\frac{n_{sl} K'_{sec,sl}}{L_{sl}} + \frac{n_{bl} K'_{sec,bl}}{L_{bl}}} \quad (9)$$

PROGRAM DESCRIPTION

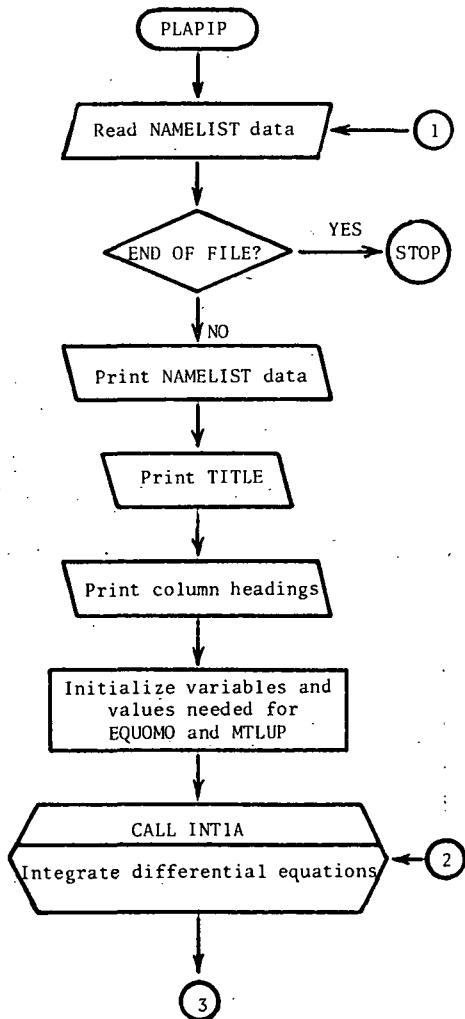
The entire computer program is written in FORTRAN IV for the Control Data 6000 series computers. The main program, PLAPIP, is used for input and output of data and initialization of variables. Program PLAPIP uses Langley Research Center library subroutine INT1A, which performs an equivalent fifth-order Runge-Kutta stepwise integration of the system differential equations. (See appendix A.) Subroutine INT1A requires a user-supplied subroutine, CHSUB, which can be used for certain logical control. For the purposes of PLAPIP, CHSUB is a blank subroutine. Subroutine INT1A also uses subroutine EQUOMO, which computes first derivatives of the system variables (\dot{v}_v , \dot{h}_v , $\dot{\gamma}$, \dot{x}_c , $d\dot{x}_c/dt$, and \dot{x}_p) at a particular time step based on tabular inputs of vehicle drag coefficient, canopy projected-area ratio (ratio of the projected area at any time to the projected

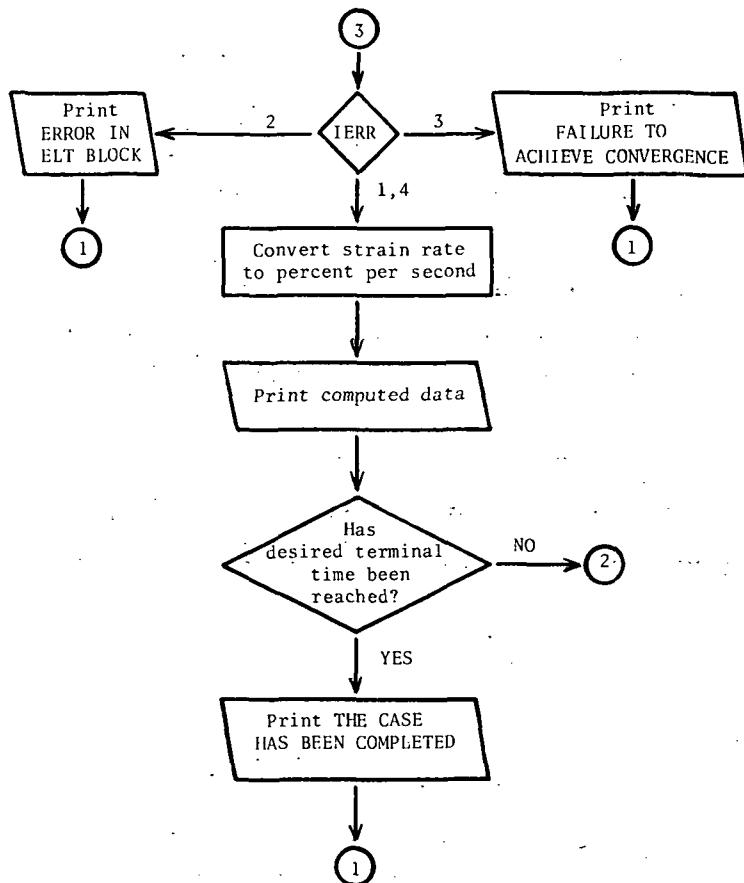
area at full inflation), canopy drag parameter (product of drag coefficient and reference area), and added mass and its rate of change. Subroutine EQUOMO uses three library subroutines: AT62 furnishes data from the U.S. Standard Atmosphere, 1962 (appendix B), MTLUP performs first- and second-order interpolation from the tabular arrays (appendix C), and DISCOT performs a first-order (or higher, if necessary) interpolation to find intermediate values of a function of two independent variables (appendix D).

Several options are available in the program. It will operate with either the U.S. Standard Atmosphere, 1962 (subroutine AT62) or user-supplied atmospheric tables. Also, the program allows for any values of viscous damping in the parachute suspension system.

Main Program PLAPIP

The main program, PLAPIP, is used for input and output of data, initialization of variables, and output of diagnostic messages. The flow diagram of program PLAPIP is as follows:





The program listing for main program PLAPIP is as follows:

```

PROGRAM PLAPIP  (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
COMMON VAR(7),CUVAR(7),DER(7),N,NT,CI,CIMAX,SPEC,ELT,ITEXT,ELE1(6)
1,ELE2(6),ALTAB(50),RHOTAB(50),VSTAB(50),NALT,XMTAB(20),CDVTAB(20),
2,NM,TINF(50),ARATIO(50),CDSTAB(50),ADMASS(50),ADMDDOT(50),NTIN,STRSL
3(50),SLMOD(50),NMOD,AMPSL(10),SLRATE(10),NSLRT,CSLTAB(100),NCSL,ST
4RBL(50),BLMOD(50),NBLMOD,AMPBL(10),BLRATE(10),NBLRT,CBLTAB(100),NC
5BL,WV,WCAN,G,OV,XLSL,XLBL,KEYAT,TLONG,DMAX,XNSL,XNBL,DRAG,ISW,QIN,
6QC,SLSTR,SLSTRR,XMIN,RADATP,XP,XPDOT,IPA,IPT,IPM,IPTL,IPTB
DIMENSION TITLE(8),ERRVAL(6)
NAMELIST/TABLE/ALTAB,RHOTAB,VSTAB,XMTAB,CDVTAB,TINF,ARATIO,CDSTAB,
1ADMASS,ADMDDOT,STRSL,SLMOD,AMPSL,SLRATE,CSLTAB,STRBL,BLMOD,AMPBL,BL
2RATE,CBLTAB,NALT,NM,NTIN,NMOD,NSLRT,NCSL,NBLMOD,NBLRT,NCBL
3/INPUT/VV0,ALTO,GAMMA0,TIME0,XCS0,XCOOT0,XP0,XPDOT0,G,DV,WV,WCAN,X
4LSL,XLBL,DMAX,RADATP,XNSL,XNBL,ISW,KEYAT,TSTOP
5/NINT/N,NT,CI,SPEC,CIMAX,ELE1,ELE2,ELT,ITEXT
EXTERNAL EQUOM0,CHSUB
PRINT 10
10 FORMAT(1H1)
20 READ(5,TABLE)
  IF (EOF,5) 998,999
998 STOP
  
```

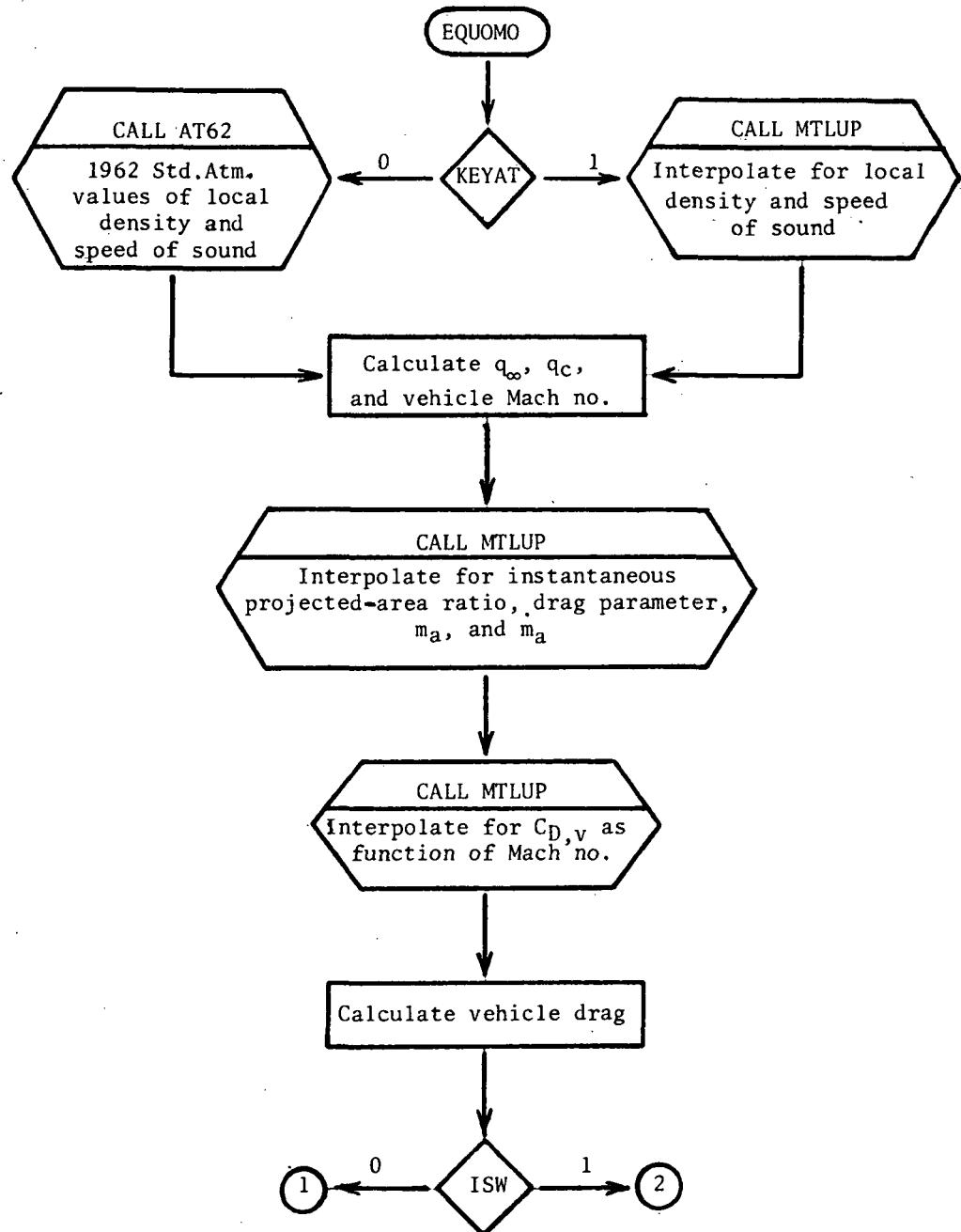
```

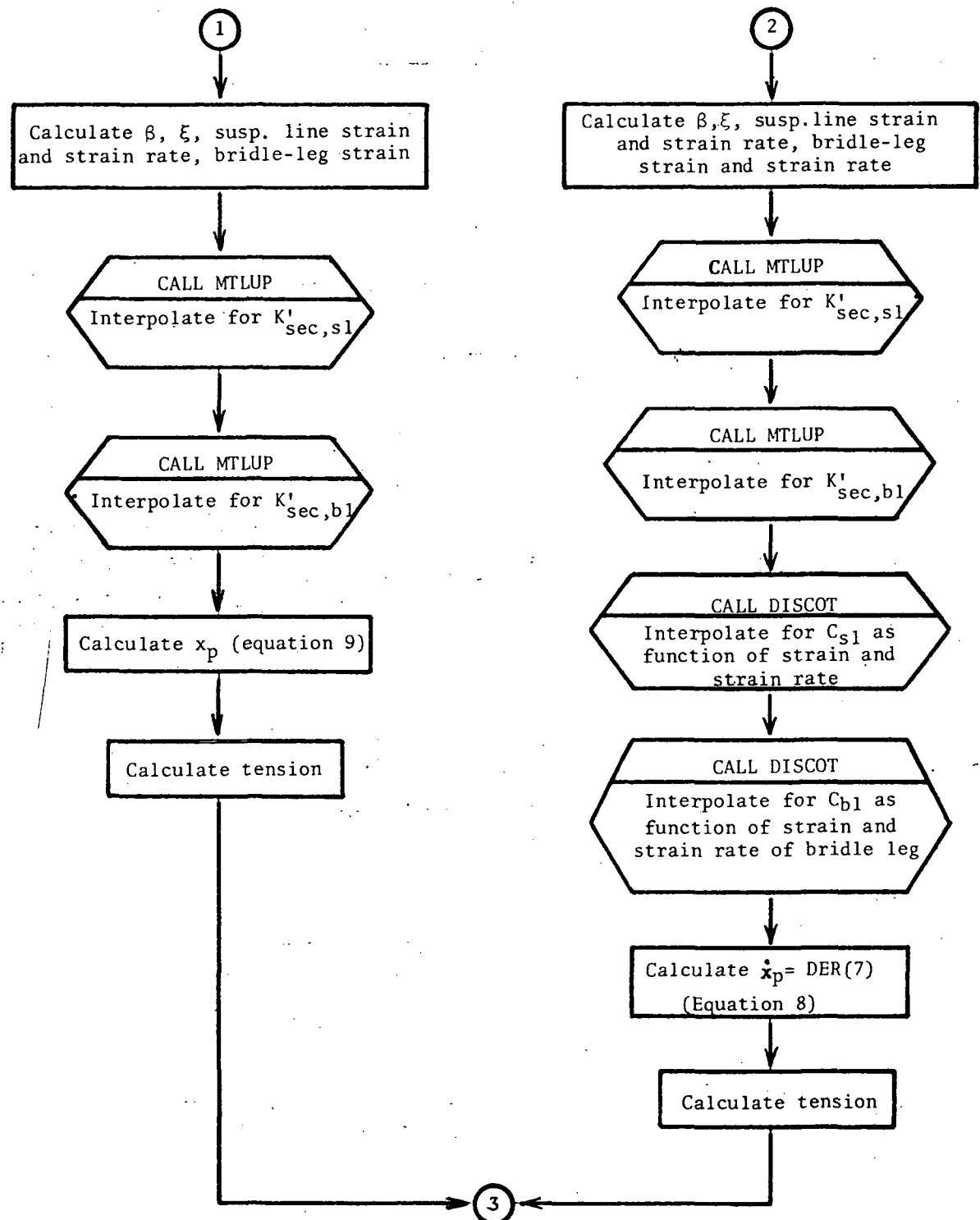
999 CONTINUE
  WRITE(6, TABLE)
  READ(5, INPUT)
  WRITE(6, INPUT)
  READ(5, NINT)
  WRITE(6, NINT)
  READ 30, TITLE
  PRINT 40, TITLE
  30 FORMAT(8A10)
  40 FORMAT(1H1,8A10)
  PRINT 50
C COLUMN TITLE FORMAT
  50 FORMAT(2X*TIME*4X*ALT*6X*VV*5X*VV-DOT*3X*M-INF*2X*Q-INF*3X*Q-CAN*4
    1X*XCS*5X*XCDOT*3X*STRSL*3X*STRRL*5X*TLONG*5X*DRAG*/2X*SEC*6X*M*7X
    2*MPS*6X*MPSS*10X*N/SM*4X*N/SM*6X*M*7X*MPS*5X*M*4X*PCNT/S*7X*N*8X
    3*N*)
C INITIALIZATION OF VARIABLES
  VAR(1)=TIME0
  VAR(2)=VVO
  VAR(3)=ALT0
  VAR(4)=GAMMA0*.017453
  VAR(5)=XCS0
  VAR(6)=XCDOT0
  VAR(7)=XP0
C INITIAL VALUES NEEDED FOR CALCULATION OF STRAIN AND STRAIN RATE IN EQUOMO
  XP=XP0
  XPDOT=XPDOT0
C INITIAL VALUES OF PARAMETERS NEEDED BY MTLUP
  IPA=-1
  IPT=-1
  IPM=-1
  IPTL=-1
  IPTB=-1
  II=0
  60 CALL INT1A(II,N,NT,CI,SPEC,CIMAX,IERR,VAR, CUVAR, DER, ELE1, ELE2, ELT,
    1ERRVAL,EQUOMO,CHSUB, ITEXT)
C ANSWERS ARE ACCEPTABLE IF IERR= 1 OR 4
  GO TO (70,80,90,70), IERR
C CONVERT STRAIN RATE TO PERCENT-PER-SECOND
  70 STRATE=SLSTRR*100.
C PRINT-OUT FORMAT FOR COMPUTED DATA
  PRINT 75, VAR(1), VAR(3), VAR(2), DER(2), XMIN, QIN, QC, VAR(5), VAR(6), SLS
  1TR, STRATE, TLONG, DRAG
  75 FORMAT(1XF5.3,1XF7.0,3XF5.0,2XF8.2,2XF5.2,2XF6.1,2XF6.1,2XF6.2,3XF
    16.2,3XF5.4,2XF7.2,4XF6.0,3XF6.0)
C CHECK TO SEE IF DESIRED TERMINATION TIME HAS BEEN REACHED
  IF(VAR(1).LT.TSTOP)GO TO 60
  PRINT 76
  76 FORMAT(/2X*THE CASE HAS BEEN COMPLETED*)
  GO TO 20
  80 PRINT 81
  81 FORMAT(/2X*ERROR IN ELT BLOCK*)
  GO TO 20
  90 PRINT 91
  91 FORMAT(/2X*FAILURE TO ACHIEVE CONVERGENCE*)
  GO TO 20
END

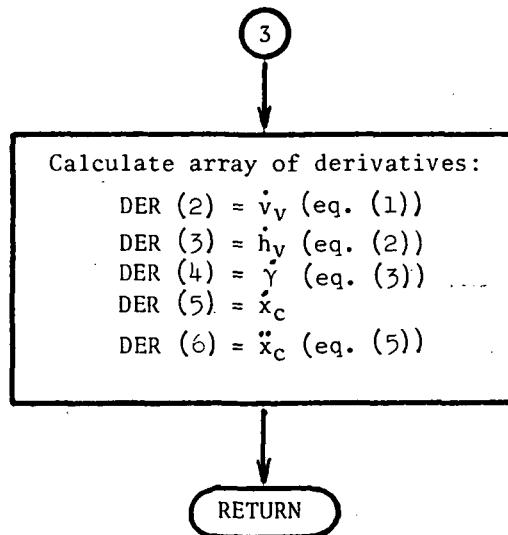
```

Subroutine EQUOMO

Subroutine EQUOMO calculates values of the first derivatives of the system variables at a particular time step. The flow diagram of subroutine EQUOMO is as follows:







The program listing for subroutine EQUOMO is as follows:

```

SUBROUTINE EQUOMO
COMMON VAR(7),CUVAR(7),DER(7),N,NT,CI,CIMAX,SPEC,ELT,ITEXT,ELE1(6)
1,ELE2(6),ALTAB(50),RHOTAB(50),VSTAB(50),NALT,XMTAB(20),CDVTAB(20),
2NM,TINF(50),ARATIO(50),CDSTAB(50),ADMASS(50),ADMDDOT(50),NTIN,STRSL
3(50),SLMOD(50),NMOD,AMPSL(10),SLRATE(10),NSLRT,CSLTAB(100),NCSL,ST
4RBL(50),BLMOD(50),NBLMOD,AMPBL(10),BLRATE(10),NBLRT,CBLTAB(100),NC
5BL,WV,WCAN,G,DV,XLSL,XLBL,KEYAT,TLONG,DMAX,XNSL,XNBL,DRAG,ISW,QIN,
6QC,SLSTR,SLSTRR,XMIN,RADATP,XP,XPDOT,IPA,IPT,IPM,IPTL,IPTB
DIMENSION ANS(4),YT(4),YA(2),VRT(50,4),VRA(50,2)
EQUIVALENCE (ARATIO,VRT),(RHOTAB,VRA)
C*****KEYAT=0 DENOTES 1962 U S STANDARD ATMOSPHERE*****
C*****KEYAT=1 DENOTES USER-SUPPLIED ATMOSPHERIC TABLES*****
IF(KEYAT.EQ.1)GO TO 10
C THESE CONVERSIONS NEEDED BECAUSE AT62 USES BRITISH UNITS
ALT=CUVAR(3)/.3048
CALL AT62(ALT,ANS)
RHO=ANS(1)*515.379
VS=ANS(4)*.3048
GO TO 15
C INTERPOLATE IN USER-SUPPLIED ATMOSPHERIC TABLES
10 CALL MTLUP(CUVAR(3),YA,1,NALT,50,2,IPA,ALTAB,VRA)
RHO=YA(1)
VS=YA(2)
15 QIN=.5*RHO*CUVAR(2)**2
QC=.5*RHO*(CUVAR(2)-CUVAR(6))**2
XMIN=CUVAR(2)/VS
C INTERPOLATE FOR AREA RATIO,CDS,ADDED MASS,AND M-DOT (YT(1) TO YT(4))
CALL MTLUP(CUVAR(1),YT,2,NTIN,50,4,IPT,TINF,VRT)
RAD=DMAX*SQRT(YT(1))/3.14159
DRAG=QC*YT(2)
CANMAS=WCAN/9.806
TOTMAS=CANMAS+YT(3)
C INTERPOLATE FOR VEHICLE DRAG COEFF. AS FUNCTION OF MACH NO.
CALL MTLUP(XMIN,CDV,1,NM,20,1,IPM,XMTAB,CDVTAB)
VDRAG=.7854*CDV*DV**2*QIN
IF(ISW.EQ.1)GO TO 20
  
```

```

C*****THIS BLOCK CALCULATES GEOMETRY, XP, AND TENSION FOR NO DAMPING*****
XSL=CUVAR(5)-XP
BETA=ATAN(RAD/XSL)
REFLNG=XSL/COS(BETA)
IF(REFLNG.LT.XSL) GO TO 25
SLSTR=REFLNG/XSL-1.
SLSTRR=CUVAR(6)/XSL/COS(BETA)
XI=ATAN(RADATP/XP)
BLSTR=XP/COS(XI)/XLBL-1.
C INTERPOLATE FOR SPECIFIC SECANT MODULI
CALL MTLUP(SLSTR,XKSL,2,NMOD,50,1,IPTL,STRSL,SLMOD)
CALL MTLUP(BLSTR,XKBL,2,NBLMOD,50,1,IPTB,STRBL,BLMOD)
A1=XNSL*XKSL
A2=XNBL*XKBL
B1=A1/XSL
B2=A2/XLBL
B3=A1*(CUVAR(5)/XSL-COS(BETA))
C CALCULATE NEW POSITION OF CONFLUENCE POINT
XP=(B3+A2*COS(XI))/(B1+B2)
TLONG=A2*(XP/XLBL-COS(XI))
DER(7)=0.
C*****END OF NO-DAMPING COMPUTATION BLOCK*****
GO TO 30
C*****THIS BLOCK CALCULATES GEOMETRY, XPDOT, AND TENSION FOR DAMPING*****
20 XSL=CUVAR(5)-CUVAR(7)
BETA=ATAN(RAD/XSL)
REFLNG=XSL/COS(BETA)
SLSTR=REFLNG/XSL-1.
IF(SLSTR.LT.0.) SLSTR=0.
XI=ATAN(RADATP/CUVAR(7))
BLSTR=CUVAR(7)/COS(XI)/XLBL-1.
IF(BLSTR.LT.0.) BLSTR=0.
SLSTRR=(CUVAR(6)-XPDOT)/XSL/COS(BETA)
BLSTRR=XPDOT/XLBL/COS(XI)
C INTERPOLATE FOR SPECIFIC SECANT MODULI
CALL MTLUP(SLSTR,XKSL,2,NMOD,50,1,IPTL,STRSL,SLMOD)
CALL MTLUP(BLSTR,XKBL,2,NBLMOD,50,1,IPTB,STRBL,BLMOD)
C DOUBLE INTERPOLATION FOR DAMPING COEFFICIENTS VS. STRAIN AND STRAIN RATE
CALL DISCOT(SLSTR,SLSTRR,AMPSL,CSLTAB,SLRATE,11,NCSL,NSLRT,CSL)
CALL DISCOT(BLSTR,BLSTRR,AMPBL,CBLTAB,BLRATE,11,NCBL,NBLRT,CBL)
A1=XNSL*CSL/XSL
A2=XNBL*CBL/XLBL
A3=XNBL*XKBL*BLSTR*COS(XI)
A4=XNSL/XSL
A5=CSL*CUVAR(6)
A6=XKSL*(CUVAR(5)-CUVAR(7)-XSL*COS(BETA))
C CALCULATE NEW VELOCITY OF CONFLUENCE POINT
C*****DER(7) IS EQUATION 8 (XP-DOT)*****
DER(7)=(A4*(A5+A6)-A3)/(A1+A2)
XPDOT=DER(7)
TLONG=COS(XI)*XNBL*(XKBL*BLSTR+CBL*XPDOT/COS(XI)/XLBL)
C*****END OF DAMPING COMPUTATION BLOCK*****
IF(TLONG.GT.0.) GO TO 30
25 TLONG=0.

```

```

C CALCULATE DERIVATIVES
C*****DER(2) IS EQUATION 1 (VV-DOT)*****
30 DER(2)=-(VDRAG+TLONG)*9.806/WV-G*SIN(CUVAR(4))
C*****DER(3) IS EQUATION 2 (HV-DOT)*****
DER(3)=CUVAR(2)*SIN(CUVAR(4))
C*****DER(4) IS EQUATION 3 (GAMMA-DOT)*****
DER(4)=-G*COS(CUVAR(4))/CUVAR(2)
C*****DER(5) IS RELATIVE VELOCITY OF CANOPY**
DER(5)=CUVAR(6)
VC=CUVAR(2)-CUVAR(6)
C*****DER(6) IS EQUATION 5 (XC-DOUBLE-DOT) ***
DER(6)=(DRAG+YT(4)*VC-TLONG+CANMAS*G*SIN(CUVAR(4)))/TOTMAS+DER(2)
RETURN
END

```

PROGRAM USAGE

The program is run on the Control Data 6000 series computers under the SCOPE 3.0 operating system and requires a field length of 40 000_g storage locations. Most cases require a central processing unit (CPU) time of 50 seconds or less. However, an exception to this rule is presented as sample case 1. This particular case involves a relatively long opening time and added mass effects and assumes no suspension-system damping.

Input Description

Input for the program is standard Control Data NAMELIST. Tables are listed under \$TABLE; initial conditions, physical-system data, and switching parameters are listed under \$INPUT; variables required by subroutine INTIA are listed under \$NINT. Input and output are in the International System of Units.

The \$TABLE input data are given as follows:

ALTAB	user-supplied array of altitude values h_v for atmospheric data, m
RHOTAB	array of atmospheric density values ρ_∞ corresponding to altitude values in ALTAB, kg/m ³
VSTAB	array of velocity of sound values corresponding to altitude values in ALTAB, m/sec
XMTAB	array of values of Mach number (see CDVTAB)
CDVTAB	array of values of vehicle drag coefficient $C_{D,v}$ as a function of XMTAB
TINF	array of values of time, sec

ARATIO	array of values of the ratio of instantaneous projected area of the canopy to projected area at full inflation as a function of TINF
CDSTAB	array of values of canopy drag parameter $(C_{DA})_c$ as a function of TINF, m^2
ADMASS	array of values of added (enclosed and apparent) mass m_a as a function of TINF, kg
ADMDOT	array of values of time rate of change of added mass \dot{m}_a as a function of TINF, kg/sec
STRSL	array of values of strain in a suspension line, m/m
SLMOD	array of values of specific secant modulus of a suspension line $K'_{sec,sl}$ as a function of STRSL, N
AMPSL	array of values of amplitude of strain in a suspension line (see CSLTAB), m/m
SLRATE	array of values of strain rate of a suspension line (see CSLTAB), $(m/m)sec^{-1}$
CSLTAB	array of values of damping coefficient of a suspension line C_{sl} as a function of both AMPSL and SLRATE, N-sec
STRBL	array of values of strain in a bridle leg, m/m
BLMOD	array of values of specific secant modulus of a bridle leg $K'_{sec,bl}$ as a function of STRBL, N
AMPBL	array of values of amplitude of strain in a bridle leg (see CBLTAB), m/m
BLRATE	array of values of strain rate of a bridle leg (see CBLTAB), $(m/m)sec^{-1}$
CBLTAB	array of values of damping coefficient of a bridle leg C_{bl} as a function of both AMPBL and BLRATE, N-sec
NALT	number of values in ALTAB array (equal to number in RHOTAB and number in VSTAB)
NM	number of values in XMTAB (and CDVTAB) array

NTIN	number of values in TINF array (also ARATIO, CDSTAB, ADMASS, ADMDOT arrays)
NMOD	number of values in STRSL (and SLMOD) array
NSLRT	number of values in SLRATE array
NCSL	number of values in CSLTAB array
NBLMOD	number of values in STRBL (and BLMOD) array
NBLRT	number of values in BLRATE array
NCBL	number of values in CBLTAB array

The \$INPUT input data are given as follows:

VV0	initial velocity of towing vehicle, v_v , m/sec
ALTO	initial altitude of towing vehicle, h_v , m
GAMMA0	initial flight-path angle of towing vehicle, γ , deg
TIME0	initial time, sec
XCS0	initial displacement of canopy skirt relative to the towing vehicle, x_{cs} , m
XCDOT0	initial velocity of canopy relative to towing vehicle, \dot{x}_c , m/sec
XP0	initial displacement of confluence point relative to towing vehicle, $x_{p,0}$, m
XPDOT0	initial velocity of confluence point relative to towing vehicle, $\dot{x}_{p,0}$, m/sec
G	local acceleration due to gravity, g , m/sec ²
DV	maximum cross-sectional diameter of towing vehicle, m
WV	weight of towing vehicle, N
WCAN	weight of parachute canopy, N

XSSL	unstressed length of a suspension line, L_{sl} , m
XLBL	unstressed length of a bridle leg, L_{bl} , m
DMAX	nominal diameter of parachute canopy, m
RADATP	radial distance from parachute central axis to point of attachment of bridle leg on towing vehicle, m
XNSL	number of suspension lines, n_{sl}
XNBL	number of bridle legs, n_{bl}
ISW	parameter denoting viscous damping mode $ISW = \begin{cases} 0 & \text{no damping} \\ 1 & \text{damping in suspension lines or bridle or both} \end{cases}$
KEYAT	parameter denoting method of determining values of atmospheric conditions $KEYAT = \begin{cases} 0 & \text{from U.S. Standard Atmosphere, 1962} \\ 1 & \text{from user-supplied tables (ALTAB, RHOTAB, VSTAB)} \end{cases}$
TSTOP	time at which user desires case to terminate, sec

The \$NINT input data are given as follows:

N	number of differential equations to be solved (for PLAPIP, $N = 6$)
NT	number of values in ELT array (for PLAPIP, $NT = 1$)
CI	initial computing interval for Runge-Kutta integration, sec
SPEC	time interval at which variable values are printed out, sec
CIMAX	maximum computing interval allowed, sec
ELE1	upper bound of local relative truncation error for the respective dependent variables

ELE2	array of "relative zeros" for the respective dependent variables (absolute values below which relative error criteria are not applied)
ELT	a specific value of time at which control is to be returned to the main program (As PLAPIP does not require such a feature, an arbitrarily large number, which is greater than TSTOP, is input.)
ITEXT	integer code allowing for printout of history of computing interval and reasons for its variations $ITEXT = \begin{cases} 0 & \text{no printout} \\ 1 & \text{printout} \end{cases}$

In addition, a title card of up to 80 characters is inserted after the final card of the \$NINT data.

Output Description

The output for program PLAPIP consists of first, the input data and then the computed data at time intervals specified by the input parameter SPEC. The input data are printed in the order in which they are read: first \$TABLE and then \$INPUT and \$NINT. The title of the case is printed immediately preceding the column headings for the computed data.

The computed data are presented columnwise as follows:

TIME	time, sec
ALT	altitude of vehicle, h_v , m
VV	velocity of vehicle, v_v , m/sec
VV-DOT	acceleration of vehicle, \dot{v}_v , m/sec ²
M-INF	free-stream Mach number
Q-INF	free-stream dynamic pressure, q_∞ , N/m ²
Q-CAN	canopy dynamic pressure, q_c , N/m ²
XCS	relative position of canopy skirt, x_{cs} , m

XCDOT	relative velocity of canopy, \dot{x}_c , m/sec
STRSL	suspension-line strain, m/m
STRRSL	suspension-line strain rate, (m/m)sec ⁻¹
TLONG	component along central axis of total tension in suspension lines T_s and bridle T_b , N
DRAG	aerodynamic drag of parachute canopy, $(C_{DA})_c q_c$, N

Sample Cases

Two sample cases are presented in order to illustrate the input and output quantities of program PLAPIP. The cases represent two inflations of a parachute, with 16.15-meter nominal diameter, in the Earth's atmosphere. One case is a subsonic inflation behind a slender towing vehicle at an altitude of 13.4 km, with the assumption of no suspension-system viscous damping. The second case is a supersonic inflation behind a bluff body at an altitude of 45 km, with suspension-system viscous damping included.

Case 1. - This case represents the inflation of a parachute, with 16.15-meter nominal diameter, behind a towing vehicle, with 0.7-meter diameter. There is no suspension-system viscous damping. Initial conditions are as follows:

$$\text{Mach number} = 0.36$$

$$h_v = 13.4 \text{ km}$$

$$q_\infty = 1406.6 \text{ N/m}^2$$

$$\gamma = -65.7^\circ$$

This case, having an opening time of 0.7 second, required a CPU time of 175 seconds.

The listing of the input data for case 1 is given as follows:


```

$INPUT
VV0 = 0.106E+03.
ALT0 = 0.134E+05.
GAMMA0 = -0.657E+02.
TIME0 = 0.11E+01.
XCS0, = 0.3008E+02.
XCDOT0 = 0.0.
XP0 = 0.21336E+01.
XPDT0, = 0.0.
G = 0.9766E+01.
OV = 0.7E+09.
WV = 0.12E+05.
WCAN = 0.205E+03.
XLSL = 0.279464E+02.
XLHL = 0.21482E+01.
DMAX = 0.1615E+02.
RADATT = 0.25E+00.
XNSL = 0.48E+02.
XNBL = 0.3E+01.
TSW = 0.
KEYAT = 0.
TSTOP = 0.19E+01.
$END
$INIT
N = 6.
NT = 1.
CI = 0.1E-01.
SPEC = 0.1E-01.
CIMAX = 0.1E-01.
ELE1 = 0.1E-03.
ELE2 = 0.1E-02.
ELT = 0.5E+03.
ITEXT = 0.
$END

```

The sample output for case 1 is given as follows:

16.15-METER PARACHUTE - SURSONIC		VV	INFLATION	NO DAMPING	DRAG
TIME	ALT	VV-DOT	Q-INF	Q-CAN	N
SEC.	M	MPSS	N/SM	N/SM	N
1.100	1340.0	106	8.55	1406.6	0.00
1.110	1339.9	106	8.54	1409.0	0.00
1.120	1339.8	106	8.53	1411.5	0.00
1.130	1339.7	106	8.50	1414.0	0.00
1.140	1339.6	106	8.45	1416.5	0.00
1.150	1339.5	106	8.37	1418.9	0.00
1.160	1339.4	107	8.25	1421.4	0.00
1.170	1339.3	107	8.07	1423.8	0.00
1.180	1339.2	107	7.80	1426.1	0.00
1.190	1339.1	107	7.27	1428.3	0.00
1.200	1339.0	107	6.59	1430.4	0.00
1.210	1338.9	107	5.19	1432.3	0.00
1.220	1338.8	107	4.47	1433.6	0.00
1.230	1338.7	107	1.14	1434.5	0.00
1.240	1338.6	107	-61	1434.7	0.00
1.250	1338.5	107	-2.29	1434.6	0.00
1.260	1338.4	107	-3.18	1434.1	0.00
1.270	1338.3	107	-2.56	1433.6	0.00
1.280	1338.2	107	-2.18	1433.0	0.00
1.290	1338.2	107	-1.55	1432.7	0.00
1.300	1338.1	107	-0.87	1432.6	0.00
1.310	1338.0	107	-0.24	1432.7	0.00
1.320	1337.9	107	-0.28	1432.8	0.00
1.330	1337.8	107	-0.39	1433.0	0.00
1.340	1337.7	107	-1.39	1433.0	0.00
1.350	1337.6	107	-3.10	1432.6	0.00
1.360	1337.5	107	-5.30	1431.9	0.00
1.370	1337.4	107	-7.70	1430.6	0.00
1.380	1337.3	107	-9.73	1428.8	0.00
1.390	1337.2	107	-11.07	1426.6	0.00
1.400	1337.1	106	-12.91	1423.9	0.00
1.410	1337.0	106	-15.13	1420.7	0.00
1.420	1336.9	106	-17.17	1416.9	0.00
1.430	1336.8	106	-18.93	1412.6	0.00
1.440	1336.7	106	-20.67	1407.9	0.00
1.450	1336.6	106	-22.57	1402.7	0.00
1.460	1336.5	105	-24.31	1397.0	0.00

1.470	13364	105	-25.78	1390.8	1048.0	31.94	13.87	0.6660	49.82
1.480	13363	105	-25.95	1384.4	1014.4	32.09	15.10	0.0715	54.26
1.490	13362	105	-27.47	1377.8	975.4	32.25	16.59	0.0777	59.64
1.500	13361	104	-29.08	1370.8	939.6	32.42	17.95	0.0844	64.55
1.510	13360	104	-30.82	1363.4	908.0	32.60	19.13	0.0916	68.81
1.520	13359	104	-33.03	1355.4	881.8	32.80	20.06	0.0992	72.18
1.530	13358	103	-34.94	1346.8	859.8	33.00	20.78	0.1075	74.81
1.540	13357	103	-36.73	1337.8	841.7	33.21	21.31	0.1160	76.73
1.550	13356	103	-38.21	1328.3	828.3	33.43	21.59	0.1241	77.78
1.560	13355	102	-40.12	1318.4	819.4	33.65	21.65	0.1324	78.00
1.570	13354	102	-42.24	1308.1	816.3	33.86	21.39	0.1406	77.10
1.580	13354	101	-44.62	1297.1	823.6	34.07	20.60	0.1483	74.26
1.590	13353	101	-47.26	1285.6	840.6	34.27	19.32	0.1556	69.66
1.600	13352	100	-50.08	1273.4	860.1	34.46	17.90	0.1625	64.54
1.610	13351	100	-53.00	1260.6	880.4	34.63	16.42	0.1689	59.23
1.620	13350	99	-55.98	1247.0	899.6	34.79	14.98	0.1747	54.03
1.630	13349	99	-59.19	1232.8	916.7	34.93	13.61	0.1802	49.12
1.640	13348	98	-61.92	1217.9	930.0	35.06	12.40	0.1851	44.75
1.650	13347	98	-64.49	1202.5	936.9	35.18	11.45	0.1896	41.34
1.660	13346	97	-66.92	1186.5	936.6	35.29	10.81	0.1938	39.06
1.670	13345	96	-69.29	1170.1	928.4	35.40	10.51	0.1977	37.98
1.680	13344	96	-71.37	1153.2	909.4	35.50	10.70	0.2017	38.66
1.690	13344	95	-73.92	1135.9	881.9	35.61	11.27	0.2057	40.73
1.700	13343	94	-76.73	1118.1	852.8	35.73	11.92	0.2099	43.07
1.710	13342	93	-79.82	1099.7	823.9	35.85	12.54	0.2144	45.35
1.720	13341	92	-83.17	1080.7	796.6	35.98	13.08	0.2190	47.30
1.730	13340	92	-87.20	1061.0	772.2	36.11	13.46	0.2238	48.67
1.740	13339	91	-90.92	1040.6	750.7	36.25	13.67	0.2286	49.44
1.750	13338	90	-94.62	1019.6	732.1	36.38	13.71	0.2334	49.58
1.760	13338	89	-98.34	997.9	716.4	36.52	13.57	0.2382	49.08
1.770	13337	88	-101.99	975.7	703.3	36.65	13.26	0.2429	47.97
1.780	13336	87	-105.28	953.0	689.3	36.78	12.98	0.2476	46.95
1.790	13335	86	-108.76	929.7	676.2	36.91	12.62	0.2519	45.65
1.800	13334	85	-112.24	906.0	668.9	37.04	11.91	0.2561	43.09
1.810	13334	83	-115.53	881.9	668.1	37.15	10.82	0.2600	39.14
1.820	13333	82	-118.49	857.5	674.2	37.25	9.33	0.2633	33.73
1.830	13332	81	-120.96	832.8	687.7	37.34	7.40	0.2661	26.77
1.840	13331	80	-122.76	808.0	709.4	37.40	5.04	0.2680	18.21
1.850	13331	79	-123.72	783.4	740.3	37.43	2.20	0.2690	7.94
1.860	13330	77	-123.66	759.0	782.1	37.44	-1.17	0.2689	-4.22
1.870	13329	76	-122.40	735.2	837.7	37.41	-5.14	0.2675	-18.57
1.880	13329	75	-119.77	712.0	911.8	37.33	-9.87	0.2646	-35.64
1.890	13328	74	-115.60	689.9	1012.4	37.21	-15.60	0.2599	-56.34
1.900	13327	73	-109.73	669.1	1154.7	37.02	-22.80	0.2529	-82.31
1.910	13327	72	-102.33	649.7	1368.8	36.74	-32.34	0.2431	-116.69

THE CASE HAS BEEN COMPLETED

Case 2. - This case represents the inflation of the same parachute behind a towing vehicle with a 3.505-meter diameter. The suspension-line viscous damping coefficient is a constant equal to 100 N-sec per line. Initial conditions are as follows:

$$\text{Mach number} = 2.16$$

$$h_v = 45 \text{ km}$$

$$q_{\infty} = 488.6 \text{ N/m}^2$$

$$\gamma = 16.0^{\circ}$$

This case, having an opening time of 0.185 second, required a CPU time of 17 seconds.

The listing of the input data for case 2 is given as follows:


```

5 INPUT
VVO = 0.705E+03.
ALT0 = 0.45E+05.
GAMA0 = 0.16E+02.
TIME0 = 0.0.
XCS0 = 0.3008E+02.
XCD0T0 = 0.0.
XPO = 0.21336E+01.
XPD0T0 = 0.0.
S = 0.967E+01.
DV = 0.3505E+01.
WV = 0.4E+04.
ACAN = 0.205E+03.
XLSL = 0.274464E+02.
XLH1 = 0.22757E+01.
DMAX = 0.1615E+02.
DADATD = 0.4E+00.
XNSL = 0.49E+02.
XNRL = 0.3E+01.
LSW = 1.
KEYAT = 0.
TSTOP = 6.25E+00.
6END)

```

```

6 INIT
V = 6.
VT = 1.
CJ = 0.5E-02.
SPEC = 0.5E-02.
CIMAX = 0.5E-02.
ELE1 = 0.1E-03.
ELE2 = 0.1E-02.
ELT = 0.5E+03.
TTEXT = 0.
8END)

```


•170	450333	791	-65.39	2.15	481.6	426.1	31.62	41.58	•0541	144.50	46317
•175	45034	701	-68.69	2.15	481.0	416.4	31.55	48.79	•0734	174.59	49158
•180	45035	701	-74.89	2.15	480.5	403.4	32.11	58.65	•0849	210.22	54471
•185	45036	700	-81.21	2.15	479.9	396.7	32.43	68.41	•0965	236.75	59896
•190	45037	700	-87.77	2.15	479.2	380.5	32.79	76.24	•1091	273.36	65526
•195	45038	699	-93.10	2.15	478.5	374.2	33.19	80.98	•1213	289.94	70108
•200	45039	699	-97.33	2.14	477.8	372.9	33.60	81.50	•1346	291.77	73740
•205	45040	699	-102.40	2.14	477.1	375.6	34.00	78.70	•1476	281.12	78093
•210	45040	699	-107.65	2.14	476.3	382.6	34.38	72.39	•1597	257.80	82607
•215	45041	697	-112.18	2.14	475.5	395.4	34.71	61.43	•1698	218.29	86503
•220	45042	697	-116.27	2.14	474.7	412.9	34.99	46.95	•1785	166.40	90012
•225	45043	696	-117.95	2.14	473.8	434.0	35.14	29.85	•1846	107.10	91467
•230	45044	696	-117.01	2.13	472.9	458.0	35.24	11.10	•1879	41.61	90680
•235	45045	695	-113.47	2.13	472.1	483.4	35.24	-3.25	•1882	-26.14	87658
•240	45046	695	-107.08	2.13	471.3	509.0	35.20	-27.26	•1850	-93.18	82197
•245	45047	694	-98.44	2.13	470.5	533.4	35.02	-44.49	•1780	-155.77	74809
•250	45048	694	-88.10	2.13	469.8	554.8	34.76	-60.12	•1694	-210.50	65960
•255	45049	693	-74.13	2.13	469.2	571.9	34.42	-72.14	•1578	-255.18	57435

THE CASE HAS BEEN COMPLETED

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., June 30, 1972.

APPENDIX A

LANGLEY LIBRARY SUBROUTINE INT1A

Language: FORTRAN

Purpose: INT1A is a closed subroutine for the solution of a set of ordinary differential equations.

Use: CALL INT1A (II, N, NT, CI, SPEC, CIMAX, IERR, VAR, CUVAR, DER, ELE1, ELE2, ELT, ERRVAL, DERSUB, CHSUB, ITEXT)

II INT1A is composed of an initialization section and an integration section. The user is required to enter the initialization section before he starts his first integration step. The above calling sequence is used for both initialization and integration with the value of the code word II determining which of the two sections of INT1A will be entered.

The user must set II = 0 in order to initialize.

During initialization the derivatives will be evaluated using the initial values of the variables but no integration will occur and control will be returned to the calling program. When INT1A is called with II > 0, entry is made to the integration section. Upon each entry to INT1A, the subroutine stores a 1 in II so that the users need not supply a value of II > 0 for repetitive integration.

Besides serving as a means for specifying the entry point to INT1A from the calling program, II can also be set to specified values in CHSUB to accomplish the following:

- 2 The user will store the integer 2 in II if the answers in CHSUB are not acceptable to him and he wishes to recompute the answers using a shorter interval. This shorter interval must be stored by the user in CI. It must be smaller than the computing interval just used.
- 3 The user will store the integer 3 in II if he wishes to return to the calling program. The answers for the interval are considered acceptable to the user and will be transferred to the VAR array (explained below) by INT1A.

In DERSUB, II may be set to:

- 4 The user will store the integer 4 in II if he wishes to discontinue calculation of the present interval and return to the calling program. On return to the calling program, the answers at the beginning of the interval will still be in the VAR array.

If the user does not set II to a value in either CHSUB or DERSUB, II will always be 1 upon the return to the calling program.

N An integer value supplied by the user which is the number of differential equations to be solved. Subroutine INT1A is compiled to solve a maximum of 20 equations but may be recompiled for larger values of N if necessary.

NT An integer value supplied by the user which is the number of values in the ELT block described below. Subroutine INT1A is compiled with a maximum of 10 values in the ELT block but may be recompiled for more values if necessary.

APPENDIX A - Continued

CI A floating-point value supplied by the user which is the computing interval INT1A will use initially. CI must be a signed value, positive if integrating forward, negative if integrating backwards. Upon entry to CHSUB, CI will contain the computing interval that INT1A will use for the next step unless it has to take a short interval to hit an ELT value or a SPEC value described below. The computing interval used on the present step is available in CHSUB as the algebraic difference between CUVAR(1) and VAR(1). Since the subroutine is used on a binary computer and the interval variation is a halving and doubling process, CI should be a power of 2.

SPEC A floating-point value supplied by the user which specifies how often he wishes INT1A to return control to the calling program so that the user may print his results.

SPEC = 0.0 Control will be returned after every acceptable integration step.

SPEC > 0.0 SPEC is the absolute value of the specified increment of the independent variable for which the user desires control returned to the calling program.

The first printout is made at the initial value of the independent variable. The next return is at the nonzero integer multiple of SPEC closest to the initial value of the independent variable. The remaining returns occur at values which have been updated from this point by the increment given in SPEC. The return times generated by the increment given in SPEC are not altered by an intervening return due to an ELT value (explained below).

CIMAX A floating-point value supplied by the user which is the absolute value of the maximum computing interval that will be used. This value will be used if the doubling process would extend the computing interval to a value larger than CIMAX. CIMAX should be set to 0.0 if there is no desired maximum.

IERR An integer value supplied by INT1A as an error code. It must be checked at every return to the calling program. It may have the following values:

- 1 A normal return, no error.
- 2 The ELT block is not monotonic in the direction of integration.
- 3 The variables have failed to meet the local truncation error requirements nine consecutive times. The answers at the beginning of the interval are still in the VAR array.
4. The variables have failed to meet the local truncation error requirements at least nine times over the last three intervals. An acceptable answer has been reached, however, and is in the VAR array.

VAR A one-dimensional array containing the independent variable followed by the N dependent variables. The user must store the N+1 initial values in the array for initialization. INT1A will store the new values of the variables in VAR after each integration step when they are accepted by the user in CHSUB. The elements of the VAR block can be printed out in the calling program in accordance with the user's specification in SPEC.

APPENDIX A – Continued

CUVAR A one-dimensional array which is given values by INT1A for two purposes. INT1A will store in the same order as the VAR array the values of the independent variable and N dependent variables at which it wishes the derivatives to be evaluated in the DERSUB subroutine. INT1A will also store the tentative answers after each integration in the CUVAR array before calling CHSUB so that the user can check these values to decide to accept or reject the answers. If accepted, the CUVAR values will then be transferred to the VAR array. No values need to be initially stored in CUVAR.

DER An N+1 array in which the user will store the derivatives evaluated in DERSUB. The derivatives should be arranged by the user in DERSUB in the same order as the VAR block so that DER(2) will be the derivative of the variable stored in VAR(2), and so forth. DER(1) will be unused. The derivatives must be computed using values of the variables which have been stored in CUVAR (not VAR) in INT1A.

ELE1 A one-dimensional array of N values supplied by the user each of which is the upper bound of local relative truncation error for the respective dependent variables. If the error for any variable exceeds its respective ELE1 value, the computing interval is halved and the integration restarted at the beginning of the present interval. If the error for all of the variables is less than 1/128 of their respective ELE1 values, the computing interval is doubled for the next integration step.

ELE2 A one-dimensional array of N values supplied by the user which represents a small value of "relative zero" for the respective dependent variables. If the absolute value of any of the variables is less than its respective ELE2 value, the relative error criteria for that variable will not be applied.

ELT A one-dimensional array of NT values supplied by the user which are values of the independent variable at which the user specifically desires control returned to his program. The values in the ELT block must be monotonic in the direction of integration or an error return will be given by INT1A.

ERRVAL A one-dimensional array of N elements in which INT1A stores an estimate of the local truncation error for each of the N dependent variables. The relative errors are computed from these values and compared with the specified ELE1 values.

DERSUB The name of a subroutine written by the user which will be called by INT1A to evaluate the derivatives. The derivatives must be stored in the DER array. INT1A will call DERSUB to evaluate the derivatives with the values of the variable it has stored in the CUVAR array. The name given to the DERSUB subroutine must appear in an EXTERNAL statement in the calling program. The user may return to the calling program by setting II to 4.

APPENDIX A – Concluded

CHSUB The name of a subroutine written by the user to allow certain logical control. After each integration step, INT1A will make available to the user in CHSUB the tentative answers in the CUVAR array. The VAR array will contain the last accepted answer (that is, the value of the variables at the beginning of the interval). Whenever the user specifies the answers are acceptable, the values in the CUVAR block are transferred to the VAR block. In CHSUB the DER block will contain the values of the derivatives evaluated with the present CUVAR block. The user has three options:

- (1) Not change II. II = 1 is considered by INT1A to denote that the user has accepted the answers in the CUVAR block. II always equals 1 upon entry to CHSUB from INT1A.
- (2) Set II = 2. The user does not accept the answers and wishes to recompute the interval using a new computing interval which he stores in CI. This computing interval must be smaller than the computing interval just used. This new value of CI will now be stored by INT1A as the normal computing interval for the subsequent integration steps.
- (3) Set II = 3. The user accepts the answer but wishes to denote a condition that he can test in the calling program. Control will be returned to the calling program with the answers in the CUVAR array transferred to the VAR array.

The name given to the CHSUB subroutine must appear in an EXTERNAL statement in the calling program.

ITEXT An integer code word supplied by the user which gives him the option to have INT1A print out a time history of the computing interval and the reasons for its variation. This printout should be requested only for problems which must be rerun because of unsatisfactory results the first time.

$$ITEXT = \begin{cases} 0 & \text{No printout requested} \\ 1 & \text{A printout requested} \end{cases}$$

Restrictions: See arguments listed under CALL statement.

Method: Subroutine INT1A, written in coordination with the other integration subroutines in the INT(x) common-usage series, is a fifth-order integration subroutine. The classical fourth-order Runge-Kutta formula is applied in conjunction with Richardson's extrapolation to the limit theory. INT1A is a variable interval size routine, in which the interval is varied to meet a specified local relative truncation error.

Accuracy: The variable interval size mode of logic is used to make available an estimate of the local relative truncation error which is then controlled as explained in the ELE1 block discussion.

Storage: 2557₈ locations.

Subroutine date: August 1, 1968.

APPENDIX B

LANGLEY LIBRARY SUBROUTINE AT62

Language: FORTRAN

Purpose: Subroutine AT62 approximates the U.S. Standard Atmosphere, 1962. It computes density in slugs/ft³, pressure in lb/ft², temperature in degrees Kelvin, and the velocity of sound in ft/sec at any geometric altitude in the range between -16 500 feet and 2 320 000 feet.

Use: CALL AT62 (Z,ANS)

Z Geometric altitude in feet

ANS A one-dimensional array that contains the results:

ANS(1) Density in slugs/ft³

ANS(2) Pressure in lb/ft²

ANS(3) Temperature in degrees Kelvin

ANS(4) Velocity of sound in ft/sec

Restrictions: For altitudes below -16 500 feet the values of density, pressure, temperature, and velocity of sound are not valid. The concept of the velocity of sound in the atmosphere becomes essentially meaningless at altitudes in excess of 300 000 feet. To point out this limitation, the velocity of sound at altitudes above 300 000 feet is set equal to the velocity of sound at 300 000 feet. For altitudes above 2 320 000 feet, density, pressure, and temperature are set equal to their respective values at 2 320 000 feet.

Method: The equations and techniques are identical to those used in computing the U.S. Standard Atmosphere, 1962.

Accuracy: The tables in reference (a) were computed with an IBM 7094 by use of some double-precision arithmetic. In converting the routine for the Control Data 6000 series computers, all double-precision arithmetic was eliminated. Accordingly, there may be slight differences between the results of the converted subroutines and the tables.

References: (a) Anon.: U.S. Standard Atmosphere, 1962. NASA, U.S. Air Force, and U.S. Weather Bur., Dec. 1962.

Storage: 1654₈ locations.

Subroutine date: August 1, 1968.

APPENDIX C

LANGLEY LIBRARY SUBROUTINE MTLUP

Language: FORTRAN

Purpose: Multiple Table Lookup (MTLUP) computes $Y_j = F_j(X)$ for $j = 1, 2, \dots, NTAB$ from a set of $NTAB$ tables using first- or second-order interpolation. An option to give Y_j a constant value for any X is also provided.

Use: CALL MTLUP (X, Y, M, N, MAX, NTAB, IP, VARI, VARD)

X	The name of the independent variable X
Y	The name of a one-dimensional array of the dependent variables calculated, $Y_j = F_j(X)$ for $j = 1, 2, \dots, NTAB$.
M	Interpolation parameter (integer), 1, 2 for first- or second-order interpolation, 0 for Y_j a constant as explained in the section entitled "Notes."
N	The number of points in each of the tables (integer), $N \leq MAX$.
MAX	The maximum number of points in each of the tables (integer) as given in the DIMENSION statement in the calling routine.
NTAB	The number of tables of dependent variables (integer).
IP	Interval pointer (integer variable name, <u>not</u> a literal integer constant), unique for a given VARI, with the following three uses: <ul style="list-style-type: none"> (1) Prior to interpolation. - If IP = 1 prior to interpolation, the values in the VARI array are checked to determine if the entire table is strictly increasing or strictly decreasing. Note that equal values in the VARI array will give an error condition. (2) For interpolation. - Upon entry, $1 \leq IP \leq N - 1$ gives the interval $(VARI(IP), VARI(IP + 1))$ to be checked first for X. (3) After interpolation. - Upon return, $0 \leq IP \leq N$ gives the interval where X was found; $VARI(IP) \leq X \leq VARI(IP + 1)$ for strictly increasing, $VARI(IP) \geq X \geq VARI(IP + 1)$ for strictly decreasing. IP = 0 indicates low-end extrapolation and IP = N indicates high-end extrapolation.
VARI	If MTLUP is used for more than one independent variable table (VARI) in a single program, a different integer variable name should be given for IP for each VARI array used. Each should be initialized to -1 and then left in variable form. (See section entitled "Method.")
VARD	The name of a two-dimensional array in which each of the NTAB columns contains the N values of a different dependent-variable (Y_j) table.

APPENDIX C - Continued

Notes: VARI(I) corresponds to VARD(I,1), VARD(I,2), . . . , and VARD(I, NTAB) for I = 1, 2, . . . , N. For M = 0 or N \leq 1, Y(J) = VARD(1,J) for J = 1, 2, . . . , NTAB, for any value of X. The program extrapolates.

Restrictions: The following arrays must be dimensioned by the calling program as indicated: VARI(MAX), VARD(MAX,NTAB), Y(NTAB).

The independent variable X and all the values in VARI and VARD must be floating point. M, N, MAX, NTAB, and IP must be integers. The values of the independent variable X in the VARI array must be strictly increasing or decreasing.

Method: The use of N \leq MAX allows the user to specify a maximum dimension (MAX) for the arrays and then run cases with smaller actual dimension (N) without recompiling, by reading N as program input.

The interval pointer IP is used to check the monotonic order of a given table only once. Thereafter, it preserves the location in the table of the previous answer.

In table lookup for scientific computing, consecutive X arguments tend to fall in the same region of independent-variable table. To take advantage of this, IP provides, for each VARI, a storage location to be used by the subroutine in communicating from one call to the next. When X is found, IP is set to the interval count where X is located. Then, on a subsequent call with the same variable name for IP (and, consequently, the same VARI table), the interval search begins with the interval that contained the previous X. This feature generally saves a significant amount of run time.

If X_1, X_2, \dots, X_N are the tabulated values of the independent variable and Y_1, Y_2, \dots, Y_N are the corresponding values of the dependent variable, then the interpolation equations (derived from references (a) and (b)) are as follows:

Second order: For i chosen such that $|X - X_i| + |X - X_{i+2}|$ is a minimum,

$$Y = \left\{ \frac{\left[Y_i(X_{i+1} - X) - Y_{i+1}(X_i - X) \right] (X_{i+2} - X)}{X_{i+1} - X_i} \right. \\ \left. - \frac{\left[Y_{i+1}(X_{i+2} - X) - Y_{i+2}(X_{i+1} - X) \right] (X_i - X)}{X_{i+2} - X_{i+1}} \right\} \frac{1}{X_{i+2} - X_i}$$

APPENDIX C – Concluded

First order: For $X_i \leq X \leq X_{i+1}$,

$$Y = Y_i + \frac{(Y_{i+1} - Y_i)(X - X_i)}{X_{i+1} - X_i}$$

Example: Given a table of 30 values of altitude (ALT) and tables of 30 values each of temperature (TEMP) and velocity (VEL) as functions of altitude,

```
DIMENSION TEMP(50), VEL(50), Y(2), ALT(50), VARD(50,2)  
EQUIVALENCE (TEMP, VARD(1,1)), (VEL, VARD(1,2))
```

IP = -1

X = 302.6

.

.

CALL MTLUP(X, Y, 1, 30, 50, 2, IP, ALT, VARD)

PRINT 10, Y

.

.

X = 198.7

CALL MTLUP(X, Y, 2, 30, 50, 2, IP, ALT, VARD)

PRINT 10, Y

.

.

Accuracy: Accuracy is a function of the order of interpolation (M) used.

References: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. Macmillan Co., c.1964, pp. 87-91.
(b) Milne, William Edmund: Numerical Calculus. Princeton Univ. Press, 1949, pp. 69-73.

Storage: 444₈ locations.

Error condition: The first call to the subroutine using a new VARI table (with IP = -1) checks the monotonic order of that table before interpolating. If an error in the order is found, the subroutine will print TABLE BELOW OUT OF ORDER FOR MTLUP AT POSITION xxx X TABLE IS STORED IN LOCATION xxxxxxx (absolute). It then prints the contents of VARI, and STOPS the program. A call with $0 \leq IP \leq N$ will interpolate but will not check the monotonic order of VARI.

If $N = M = 2$, the "TABLE BELOW . . ." printing is given and execution STOPS. In this case, "AT POSITION xxx" has no relevance.

Subroutine date: September 12, 1969.

APPENDIX D
LANGLEY LIBRARY SUBROUTINE DISCOT

Language: FORTRAN

Purpose: DISCOT performs single or double interpolation for continuous or discontinuous functions.

Given a table of some function y with two independent variables, x and z , this subroutine performs K_x th- and K_z th-order interpolation to calculate the dependent variable. In this subroutine all single-line functions are read in as two separate arrays and all multiline functions are read in as three separate arrays; that is,

$$\begin{aligned} x_i & \quad (i = 1, 2, \dots, L) \\ y_j & \quad (j = 1, 2, \dots, M) \\ z_k & \quad (k = 1, 2, \dots, N) \end{aligned}$$

Use: CALL DISCOT (XA, ZA, TABX, TABY, TABZ, NC, NY, NZ, ANS)

XA	The x argument
ZA	The z argument (may be the same name as x on single lines)
TABX	A one-dimensional array of x values
TABY	A one-dimensional array of y values
TABZ	A one-dimensional array of z values
NC	A control word that consists of a sign (+ or -) and three digits. The control word is formed as follows: (1) If $NX = NY$, the sign is negative. If $NX \neq NY$, then NX is computed by DISCOT as $NX = NY/Nz$, and the sign is positive and may be omitted if desired. (2) A one in the hundreds position of the word indicates that no extrapolation occurs above z_{max} . With a zero in this position, extrapolation occurs when $z > z_{max}$. The zero may be omitted if desired. (3) A digit (1 to 7) in the tens position of the word indicates the order of interpolation in the x -direction. (4) A digit (1 to 7) in the units position of the word indicates the order of interpolation in the z -direction.
NY	The number of points in y array
NZ	The number of points in z array
ANS	The dependent variable y

APPENDIX D – Continued

The following programs will illustrate various ways to use DISCOT:

CASE I: Given $y = f(x)$

NY = 50

NX (number of points in x array) = NY

Extrapolation when $z > z_{\max}$

Second-order interpolation in x-direction

No interpolation in z-direction

Control word = -020

DIMENSION TABX (50), TABY (50)

1 FORMAT (8E 9.5)

READ (5,1) TABX, TABY

READ (5,1) XA

CALL DISCOT (XA, XA, TABX, TABY, -020, 50, 0, ANS)

CASE II: Given $y = f(x,z)$

NY = 800

NZ = 10

NX = NY/NZ (computed by DISCOT)

Extrapolation when $z > z_{\max}$

Linear interpolation in x-direction

Linear interpolation in z-direction

Control word = 11

DIMENSION TABX (800), TABY (800), TABZ (10)

1 FORMAT (8E 9.5)

READ (5,1) TABX, TABY, TABZ

READ (5,1) XA, ZA

CALL DISCOT (XA, ZA, TABX, TABY, TABZ, 11, 800, 10, ANS)

CASE III: Given $y = f(x,z)$

NY = 800

NZ = 10

NX = NY

Extrapolation when $z > z_{\max}$

Seventh-order interpolation in x-direction

Third-order interpolation in z-direction

Control word = -73

DIMENSION TABX (800), TABY (800), TABZ (10)

1 FORMAT (8E 9.5)

READ (5,1) TABX, TABY, TABZ

READ (5,1) XA, ZA

CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -73, 800, 10, ANS)

CASE IV: Same as Case III with no extrapolation above z_{\max} . Control word = -173

CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -173, 800, 10, ANS)

APPENDIX D - Continued

Restrictions: See rule (5c) of section "Method" for restrictions on tabulating arrays and discontinuous functions. The order of interpolation in the x- and z-directions may be from 1 to 7. The following subprograms are used by DISCOT: UNS, DISSER, LAGRAN.

Method: Lagrange's interpolation formula is used in both the x- and z-directions for interpolation. This method is explained in detail in reference (a) of this subroutine. For a search in either the x- or z-direction, the following rules are observed:

(1) If $x < x_1$, the routine chooses the following points for extrapolation:

$$x_1, x_2, \dots, x_{k+1} \text{ and } y_1, y_2, \dots, y_{k+1}$$

(2) If $x > x_n$, the routine chooses the following points for extrapolation:

$$x_{n-k}, x_{n-k+1}, \dots, x_n \text{ and } y_{n-k}, y_{n-k+1}, \dots, y_n$$

(3) If $x \leq x_n$, the routine chooses the following points for interpolation:

When k is odd,

$$x_{i-\frac{k+1}{2}}, x_{i-\frac{k+1}{2}+1}, \dots, x_{i-\frac{k+1}{2}+k} \text{ and } y_{i-\frac{k+1}{2}}, y_{i-\frac{k+1}{2}+1}, \dots, y_{i-\frac{k+1}{2}+k}$$

When k is even,

$$x_{i-\frac{k}{2}}, x_{i-\frac{k}{2}+1}, \dots, x_{i-\frac{k}{2}+k} \text{ and } y_{i-\frac{k}{2}}, y_{i-\frac{k}{2}+1}, \dots, y_{i-\frac{k}{2}+k}$$

(4) If any of the subscripts in rule (3) become negative or greater than n (number of points), rules (1) and (2) apply. When discontinuous functions are tabulated, the independent variable at the point of discontinuity is repeated.

(5) The subroutine will automatically examine the points selected before interpolation and if there is a discontinuity, the following rules apply. Let x_d and x_{d+1} be the point of discontinuity.

(a) If $x \leq x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d-k}, x_{d-k+1}, \dots, x_d \text{ and } y_{d-k}, y_{d-k+1}, \dots, y_d$$

(b) If $x > x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d+1}, x_{d+2}, \dots, x_{d+k} \text{ and } y_{d+1}, y_{d+2}, \dots, y_{d+k}$$

(c) When tabulating discontinuous functions, there must always be $k+1$ points above and below the discontinuity in order to get proper interpolation.

(6) When tabulating arrays for this subroutine, both independent variables must be in ascending order;

APPENDIX D - Concluded

(7) In some engineering programs with many tables, it is quite desirable to read in one array of x values that could be used for all lines of a multiline function or different functions. Even though this situation is not always applicable, the subroutine has been written to handle it. This procedure not only saves much time in preparing tabular data, but also can save many locations previously used when every y coordinate had to have a corresponding x coordinate. Another additional feature that may be useful is the possibility of a multiline function with no extrapolation above the top line.

Accuracy: A function of the order of interpolation used.

Reference: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956.

Storage: 555_8 locations.

Subprograms used: UNS 40_8 locations.

DISSER 110_8 locations.

LAGRAN 55_8 locations.

Subroutine date: August 1, 1968.

REFERENCES

1. Preissner, John S.; and Greene, George C.: Effect of Suspension Line Elasticity on Parachute Loads. *J. Spacecraft & Rockets*, vol. 7, no. 10, Oct. 1970, pp. 1278-1280.
2. Poole, Lamont R.; and Huckins, Earle K., III: Evaluation of Massless-Spring Modeling of Suspension-Line Elasticity During the Parachute Unfurling Process. *NASA TN D-6671*, 1972.
3. Amer. Power Jet Co.: Performance of and Design Criteria for Deployable Aerodynamic Decelerators. *ASD-TR-61-579*, U.S. Air Force, Dec. 1963. (Available from DDC as AD 429 921.)

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE \$300

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION



NASA 451

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546